Sharing Uncertain Graphs with Syntactic Anonymity

Dongqing Xiao, Mohamed Y. Eltabakh, Xiangnan Kong
Computer Science Department, Worcester Polytechnic Institute
MA, United States of America
{dxiao, meltabakh, xkong} @wpi.edu

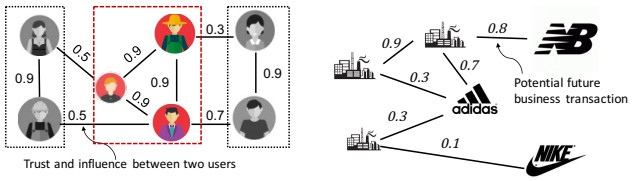
ABSTRACT

Many graphs in real-world applications, such as social network and business to business network, are not deterministic but are uncertain. Related research such as social science and viral marketing requires open access to such uncertain graph datasets. While sharing these datasets often risks exposing sensitive data to the public. Current works mainly concern about privacy issues with deterministic graph sharing. The uncertain scenario is overlooked.

In this paper, we study the problem of sharing uncertain graphs with syntactic anonymity. We first show conventional methods are not applicable for sharing uncertain graph. By disregarding the possible world semantic of uncertain graphs, they significantly disrupt the stochastic structure. Our work seeks a solution to share meaningful probabilistic graphs without compromising user privacy. We develop a syntactically private algorithm, Squid, for solving this problem. It integrates the possible world semantic into the core of anonymization. It enables a fine-grained, uncertainty-aware control over the injected noise. We apply our method to real uncertain graphs and show its efficiency and practical utility.

1. INTRODUCTION

Graphs are widely used to capture the relationships in emerging applications, such as business to business (B2B) and social networks. Sometimes, the existence of the relationship between two entities is uncertain (probabilistic). For instance, in social networks, nodes represent individual users, while edges represent friendship or trust link among them. Usually, the link strength is derived by inference and prediction models built on interaction details [1, 17]. While edge probability denotes the accuracy of a link prediction or the trust of one person on another. In these applications, the data can be modeled and shared as uncertain graphs whose edges carries a probability of existence. The probability represents the confidence that the relationship holds in reality.



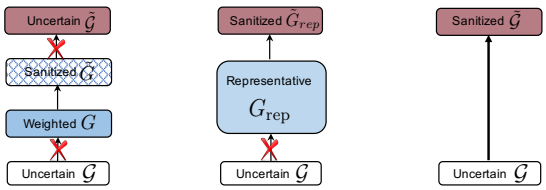
(a) Social Trust Network (b) B2B Network

Figure 1: Real-world uncertain graphs with privacy concerns.

These uncertain graphs are invaluable for scientific research and commercial applications e.g., understanding social interactions, information propagation and advertising [7, 17]. Compared to sharing the results of graph mining, graph sharing gives greater flexibility as recipients can perform unlimited analysis, data explosion with novel methods.

However, sharing these uncertain graphs could seriously jeopardize the privacy of users or entities profiled inside. In social trust network, the trust relationships among users, which significantly impact users' behaviors, are usually probabilistic. They are useful in social interaction study and micro-targeting [17]. However, users are unwilling to share such confidential information with potential adversaries. In B2B networks, business operators also hesitate to share transaction patterns as it relates to confidential business models. Such tension is raising the question of sharing uncertain graphs without compromising privacy.

A number of privacy preserving graph sharing schemes have been studied in the deterministic scenario [18–21, 26, 31, 33, 35], though many problems still remain unexplored in the uncertain scenario.



(a) Casting-based (b) Rep-based (c) Squid

Figure 2: Illustration of three anonymization approaches.

An obvious approach is to convert the uncertain graph sharing problem into the deterministic one by casting edge probabilities as edge weights. While, the general idea is appealing, the casting-based method is inadequate to provide desirable utility for uncertain graph. By disregarding the

possible world semantics of the uncertain graph, casting-based approach, as later illustrated, fails to reflect uncertain graph properties such as connectivity, dense subgraphs correctly [15, 36]. Hence, the casting-based scheme could produce poor results in the uncertain scenario even if the weighted graph anonymization algorithm is good.

Example. *Connectivity of deterministic subgraphs is generally measured by the concept of cut, which is defined as the sum of weights of intra edges. Generally, the bigger the cut, the harder to separate two subgraphs. In Figure 1(a), the equal cut* $C(SG_1, SG_2) = C(SG_3, SG_2) = 1$ *implies the identical connectivity of* SG_1 *and* SG_3 *w.r.t.* SG_2*. However, with the possible world semantics, we know the probability to separate* SG_1 *and* SG_2 *is* $(1-0.5)^2 = 0.25$*, and that to separate* SG_2 *and* SG_3 *is* $(1-0.3)(1-0.7) = 0.21$*. Hence, in fact,* SG_2 *is closer to* SG_1 *than to* SG_3*.*

In the previous work [11], we ever proposed a representative-based method, based on the idea of processing uncertain graph through representative instances [23]. It first extracts a single deterministic representative instance G that capture structural properties of the uncertain graph. After that, anonymization can then proceed efficiently on G using conventional algorithms, regardless of the uncertainty. The advantage of this method is that it does not require new anonymization techniques. However, the representative-based method is not always feasible. The detachment of edge uncertainty deteriorates the data utility.

As ever discussed, conventional graph anonymization schemes are inadequate to share uncertain graphs with a desirable trade-off between privacy and utility. It is worthwhile to consider developing the specially optimized solution for handling following challenges.

• *Stochastic Privacy Attacks.* Edge uncertainty plays an indispensable role in the uncertain graph model. Discarding them in the release is impractical. However, the extra release of edge uncertainty makes privacy protection far more difficult as it empowers the adversary and makes the profiled entity more vulnerable. To this end, we clarify the potential re-identification attack and adopt a privacy notion for releasing privacy-preserving uncertain graphs.

• *Stochastic Utility Loss Metric.* It is challenging to maintain the structure when the uncertain graph is modified to pursue anonymity. The structural distortion incurred is evaluated by the specially designed utility loss metric. It aims to safeguard the utility of the release graphs. Unfortunately, current graph utility loss metrics such as graph edit distance [19], spectrum discrepancy [35], community reconstruction error [31] and shortest path discrepancy [20] are not suitable in our problem setting because of the ignorance of edge uncertainty. In this context, the discrepancy w.r.t the uncertain graph reliability becomes a useful criterion. It evaluates the connectivity difference in the context of the entire graph and meanwhile utilizes the possible world semantics.

• *Intractable Search Space.* Informally, our goal is to find a sanitized graph with the desired level of privacy with as few graph mutations as possible. Even the simple deterministic graph anonymization problem, *i.e.*, is known to be NP-hard when mutations are limited to edge additions and deletions [12]. In the probabilistic scenario, edge modifications are no longer limited to edge addition and deletion, but can be infinite probability deviations. Exhaustive search is computationally intractable if the number of edges is large. It makes the uncertain graph anonymization problem very challenging. In this work, we approximate the problem of interest via a randomized algorithm, which built on the basis of meta-heuristics.

In this work, we propose a novel sanitization solution tailored towards uncertain graphs via incorporating possible world semantics and test it on three real-life network datasets. We analyze the utility of sanitized graphs (uncertain ones) in two scenarios: extracting statistics for graph analysis, and performing influence prorogation. We show our method preserves as much the stochastic nature of the original uncertain graph as possible while injecting necessary structural noise to guarantee a desirable level of privacy. We also show that significant improvement of our methods over casting-based and representative-based methods. Specifically, we make the following contributions.

- We are the first to formulate the uncertain graph sharing problem. We show the potential re-identification attack and present a practical privacy notion.
- Motivated by the use of connectivity error, we propose a utility loss metric on the basis of reliability. It evaluates the connectivity difference in the context of the entire graph and also utilizes the possible world model.
- To alleviate the combinational intractability, we propose a randomized algorithm boosted by the hybrid of uncertainty-aware heuristics. It excels in identifying a population of sanitized results with good quality efficiently.
- We conduct extensive experimental studies to demonstrate efficiency and practical utility of our algorithms. The results demonstrate a significant advantage over the conventional methods that do not directly consider edge uncertainties.

The rest of the paper is organized as follows. In Section 2, we summarize related works, point out the limitation of existing methods, and clarify our distinct privacy goal. In Section 3 we formulate the uncertain graph-anonymization problem. Sections 4 5 present our anonymization approach for releasing privacy-preserving uncertain graph. In Section 6 we apply our method to several real-world uncertain graphs and demonstrate its performance, practical utility and efficiency.

2. RELATED WORK

A significant amount of prior work has been done on protecting the privacy of network datasets. The comprehensive survey is out of the scope of this paper. Here, we briefly summarize related work and clarify our privacy goal.

Syntactic Privacy. Early works on privacy-preserving network releasing focus on developing anonymization techniques. Many of them modify the graph structure in subtle ways that guarantee privacy but keep much of graph structure for release. The released graph is available for all the analysis tasks. These approaches usually provide privacy protection against specific de-anonymization attacks. Most of them employ syntactic privacy models derived from k-anonymity [28] which requires creating k same entities (*e.g.* neighborhoods, degree nodes) to blend victims.

Related anonymization methods can be classified into four main categories: (1) Clustering-based generalization [3, 13,

14]; (2) *Edge modification* [19,27,31,32,37,38]; (3) *Edge randomization* [20, 22, 35]; and (4) *Uncertainty semantic-based modifications* which add uncertainty to some edges and thus converting the deterministic graph to an uncertain version for anonymity [4, 21].

In the first category, Hay *et al.* [14] proposed to generalize a network by clustering nodes and only publish the hyper-graph (# of nodes in each partition with # of edges within and across partitions). Campan *et al.* [2] studied the attributed graph case with a similar solution. Cohen *et al.* [29] presented a sequential clustering algorithm with better utility preserving. While, Cormode*et al.* [3] payed attention to attributed-based matching attack. To this end, their method marks the mapping by clustering the nodes and corresponding real world entities into groups.

In the second category, Liu *et al.* [19] focused on resisting degree-based entity re-identification attacks. They propose to add and delete edges to pursue k-degree anonymity. Zhou *et al.* [37] consider stronger re-identification attack based on radius-one subgraph. Zhou *et al.* [38] assume that the adversary knows the compute graph. Their algorithms use edge addition and deletion to make graph k-Automorphism.

In the third category, Hay *et al.* [20] study the use of random perturbation for identity obfuscation. They consider the basic degree-based re-identification of nodes. Besides, they propose to quantify the level of anonymity that is provided for the given node v in the real network by the perturbed graph as the inverse of the maximum of the belief probability $Pr(v|u)$. Ying *et al.* [35] compare random perturbation methods to the method of $k-$degree anonymity. Their experiments over two datasets (Enron and Pollblogs) show the deterministic edge modification methods for k-degree anonymity preserve the graph structure better than random perturbation methods.

The uncertainty semantic-based approaches are known as the state-of-art ones because of their excellent privacy-utility trade-off, brought by the fine-grained perturbation leveraging the uncertain semantics. Our method belongs to the fourth category while for the probabilistic context.

Differential Privacy. The dependence of adversary knowledge makes graph anonymization methods are vulnerable to attackers with strong background knowledge than assumed. Such fact has similuated the use of differential privacy for more rigorous privacy guarntess. The recent research on applying differential privacy to graph data roughly falls into two directions. The first direction aims to release specific differentially private mining results, such as degree distributions, sub-graph counts, and frequent graph patterns [10, 33]. These methods only publish query result. However, there are many situations in which answering statistical queries simply does not achieve the purpose of sharing the graph. The second direction aims to share the meaningful graph. Most research in this direction [25,26] projects an input graph to dK-series and ensures differential privacy on dK-series statistics. Later, private statistics are then either fed into generators or MCMC process to generate a fit synthetic graphs. While current techniques are still inadequate to provide desirable data utility for many graph mining tasks. Wang *et al.* [30] propose to project a graph to the spectral domain and inject noise to the eigenvalues and eigenvector. This approch achieved significant improvement on efficiency, which, is still not able to achieve good data utility. Xiao *et al.* [33] present a solution based on structural inference over the hierachical random graph model. This approach achieved the reasonable utility over real-life graph datasets.

All the methods target at providing privacy guarantee to the deterministic graph. The uncertain scenario is unexplored.

2.1 Our Privacy Goal

As ever discussed, existing methods fail to provide utility guarantee in the uncertain scenario. In this work, we try to move this line of research one step forward from the deterministic context to a broader probabilistic context. Our work seeks a solution to share meaningful uncertain graphs while preserving privacy. We believe the anonymization process needed to be specially optimized to uncertain graphs.

There is a widespread belief that differential privacy and its offsprings are immune to various privacy attacks. It offers a guarantee bound ϵ on the loss of privacy due to the data release [26,33]. However, there is no clear way to set general policy for choosing the privacy parameter ϵ for sufficient privacy guarantee [18]. Its implications and impacts on the risk of disclosure in practice heavy depend on data detail. Thus, differential privacy is difficult to apply in practice.

In contrast, the notion of syntactic privacy can be defined and understood based on the data schema. And, its parameters have a clear privacy meaning that can be understood independent of the actual data. Moreover, they have a clear relationship to the privacy regulation of individual identifiability (e.g., General Data Protection Regulation GDPR). Hence, we focus on sharing uncertain graphs with syntactic anonymity.

3. PROBLEM FORMULATION

In this section, we present background on uncertain graphs, show possible privacy attacks, and formulate the privacy notion for releasing privacy-preserving uncertain graphs. We also justify our choice of utility loss metric. Finally, we present our formulation of the uncertain graph anonymization problem.

3.1 Uncertain Graph

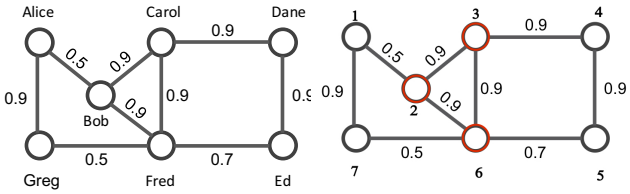
An uncertain graph $\mathcal{G} = (V, E, p)$, is defined over a set of nodes V, a set of edges E, and a set of probabilities p of edge existence. Following the literature [8, 24, 36], we assume the possible-worlds semantics, and we consider the edge probabilities independent [1]. An uncertain graph $\mathcal{G} = (V, E, p)$ essentially represents a probability distribution over all of the certain graphs G in the forms of which the uncertain graph may actually exist. The probability of observing any possible world $G_i = (V, E_{G_i})$ is

$$Pr[G_i] = \prod_{e \in E_{G_i}} p(e) \prod_{e \in E \setminus E_{G_i}} 1 - p(e)$$

3.2 Privacy Attack

Apparently, simply removing the identities of the nodes before publishing the uncertain graph does not guarantee privacy. The structure of the uncertain graph itself, and in its basic form the degree of the nodes, can be revealing the identities of individuals. In practice, the adversary may

[1]We leave the conditional probability model as a future extension.



(a) Social Trust Network (b) The naive anonymization

Figure 3: The structural re-identification issue.

have access to external information about the entities in the graphs. This information may be obtained by the adversary's malicious actions.

Example. For the uncertain graph in Figure 3, the adversary might know that "*Fred has **three or more** trust neighbors*". Such information allows the adversary to narrow down the set of candidates in the sanitized graphs. The statement partially re-identify Fred as $\{2, 3, 6\}$ with different **probabilities** respectively. Different to the deterministic scenario, such posterior probabilities significantly vary over candidate nodes $\{2, 3, 6\}$ where $P(\text{Fred}|2) \ll P(\text{Fred}|6) \ll P(\text{Fred}|2)$ as $P(\text{Fred}|2) = 0.5{*}0.9{*}0.9 = 0.405$ and $P(\text{Freq}|2) = 0.9 * 0.9 * 0.9 = 0.729$ and $P(\text{Freq}|6) = 0.468$.

As ever illustrated, nodes in uncertain graphs are vulnerable to the re-identification risk. Such Entity Re-identification (ER) can lead to additional disclosures. In this work, we focus on the ER attack as it is one of the most serious privacy problems.

3.3 Privacy Notion

In this work, we adopt (k, ϵ)-obf, a variant of the well-known $k-$anonymity as the privacy notion for privacy-preserving uncertain graph releasing. It was proposed by Boldi *et al.* in [4], where $k \geq 1$ is a desired level of obfuscation and $\epsilon \geq 0$ is a tolerance parameter.

Obfuscation Parameter Similar to k-candidate anonymity, $k-$obf requires blending every entity with other fuzzy matching entities. While, k-anonymity requires that each entity is identical with at least k-1 other entities. Bonchi*et al.* [6] observed that k-anonymity does not measure the amount of uncertainty correctly that the adversary has regarding the correct identification of the target individual.

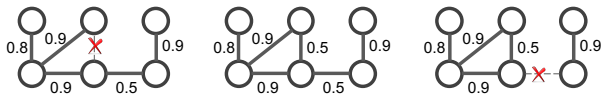
k-obf generalizes the candidate anonymity concept to the probabilistic scenario by the use of entropy. For a given node v in the real network, it quantifies the level of obfuscation that is provided for v by the perturbed graphs as the entropy of $\{Pr(v|u) : u \in V_{\tilde{G}}\}$, where $Pr(v|u)$ stands for the posterior belief probability that u is the image of the target node v.

Though it is initially used to measure the anonymity provided by an uncertain graph to the deterministic graph, the stochastic nature makes it a good fit in the uncertain scenario.

Tolerance parameter As for the tolerance parameter ϵ, it serves for the following purpose. There might be extreme unique nodes, e.g., Trump in a Twitter network, whose obfuscation is almost impossible. Thus, Boldi *et al.* [4] introduce a tolerance parameter ϵ, which allows skipping up to $\epsilon{*}|V|$ nodes and makes the privacy notion more practical.

3.4 Utility Loss: Reliability Discrepancy

In the context of deterministic graphs, connectivity discrepancy is widely used to measure the structural distortion for the following reasons. First, the connectivity model is known to yield a better graph representation than the degree sequence model. What's more, connectivity plays a vital role in various graph mining tasks such as nearest neighbor locating, decomposition and graph clustering.



A perturbation $\hat{G}_1$ A graph G A perturbation $\hat{G}_2$

Graph Edit Distance (GED): $GED(\hat{G}_1, G) = GED(\hat{G}_2, G)$

Reliability Discrepancy(RD): $RD(\hat{G}_1, G) \ll RD(\hat{G}_2, G)$

Figure 4: Comparison of utility loss metrics.

While, the concept of reliability generalizes the connectivity concept in the uncertain scenario. It captures the probability that two given nodes are reachable over all possible worlds, as shown in Def 1. Analogous to the deterministic case, we use reliability discrepancy as the utility-loss metric for utility safeguard, as outlined in Def 2.

Definition 1. ***Two-Terminal Reliability [8]*** *Given an uncertain graph $\mathcal{G}$, and two distinct nodes u and v in the graph, the reliability of (u, v) is defined as:*

$$R_{u,v}(\mathcal{G}) = \sum_{G \in W(\mathcal{G})} \mathcal{I}_G(u, v)\ Pr[G]$$

where $Pr[G]$ is the probability of observing G as one possible world of G, and $\mathcal{I}_G(u, v)$ is 1 iff u and v are contained in a connected component in G, and 0 otherwise.

Definition 2. ***Reliability Discrepancy (RD)*** *The reliability difference between a sanitized output $\tilde{\mathcal{G}}$ and the original input $\mathcal{G}$, denoted as $\Delta(\tilde{\mathcal{G}})$, is defined as the sum of the two-terminal reliability discrepancy over all node pairs $(u, v) \in V_{\mathcal{G}}$.*

$$\Delta(\tilde{\mathcal{G}}) = \sum_{(u,v) \in V_{\mathcal{G}}} |R_{u,v}(\mathcal{G}) - R_{u,v}(\tilde{\mathcal{G}})|$$

3.5 Problem Statement

Problem 1. *Given an uncertain graph $\mathcal{G}$ and desired anonymization parameters k and ϵ, the objective is to find a (k, ϵ)-obf uncertain graph $\tilde{\mathcal{G}}$ with the minimal utility loss,*

$$\begin{aligned} &\underset{\tilde{\mathcal{G}}}{\text{argmin}} && \Delta(\tilde{\mathcal{G}}) \\ &\textit{Subject to} && \tilde{\mathcal{G}}\ \textit{is}\ (k, \epsilon) - \textit{obf} \end{aligned}$$

4. THE STATE-OF-ART APPROACH

Before presenting our solution Squid, we first describe the state-of-art graph anonymization approach (Obf) [4] in the deterministic scenario. The purpose of describing it is to separate the basic framework with the contribution of Squid. They differ in the search strategy of sanitized candidates.

4.1 Overview

The Obf method obfuscates the (deterministic) graph data by adding or removing edges *partially*. For each edge e, it assigns a probabilistic deviation $r_e \in [0, 1]$, where $r_e \leftarrow R(\sigma)$. In particular, the uncertainty injecting scheme proceeds as follows:

$$p(e) = \begin{cases} 1 - r_e & e \in E \\ r_e & otherwise \end{cases} \quad (1)$$

Generally, it transfers edge existence from existing edges to non-existing ones for identification obfuscation.

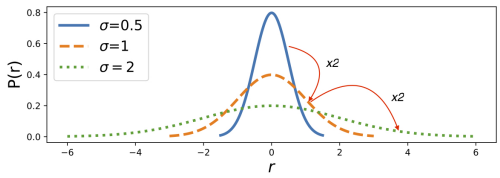


Figure 5: Illustration of the obfuscation effect brought by larger values of standard deviation σ.

For the high utility of the obfuscated graph, smaller values of the parameter r_e should be favored. The most widely known member of generating distribution $R(\sigma)$ is the truncated normal distribution with mean 0 and variance σ^2. In principle, R could be any distribution. As the standard deviation σ decreases, a greater mass of R_σ will concentrate near $r_e = 0$. As illustrated in Figure 5, smaller values of σ generally contributes towards better utility preserving, while at the same time they provide lower level of obfuscation. Larger values of σ have the opposite effect. Then, the amount of injected noise and consequent structural distortion will be smaller. Aiming at the high utility, Obf aims at injecting the minimal amount of uncertainty need to achieve the necessary obfuscation. As outlined in Algo 1, it computes the minimal amount of uncertainty via a binary search on the σ value.

Algorithm 1 The obfuscation algorithm

Input: Graph $\mathcal{G}$, obfuscation level k, tolerance parameter ϵ
Output: The result $\mathcal{G}_{obf}$

$\sigma_l \leftarrow 0$; $\sigma_u \leftarrow 1$
repeat
  $\langle \hat{\epsilon}, \hat{\mathcal{G}} \rangle \leftarrow$ **genObf**(-,σ_u)
  if $\hat{\epsilon} = 1$ (fail) **then** $\sigma_l \leftarrow \sigma_u$; $\sigma_u \leftarrow 2\sigma_u$
until $\hat{\epsilon} \neq 1$
repeat
  $\sigma_{mid} \leftarrow (\sigma_u + \sigma_l)/2$
  $\langle \hat{\epsilon}, \hat{\mathcal{G}} \rangle \leftarrow$ **genObf**(-,σ_{mid})
  if $\hat{\epsilon} = 1$ **then** $\sigma_l \leftarrow \sigma_{mid}$
  else $\sigma_u \leftarrow \sigma_{mid}$; $\mathcal{G}_{obf} \leftarrow \hat{\mathcal{G}}$
until $\sigma_u - \sigma_l$ is enough small
return $\mathcal{G}_{obf}$

The function `genObf` determines the search flow. The function `genObf` handles the search of (k, ϵ)-*obf* instances using a given standard deviation parameter σ. It either returns the found sanitized instance or failure sign. The search starts with an initial guess of an upper bound σ_u, which is iteratively doubled until a (k, ϵ)-*obf* instance is found. Then, the binary search is performed over the range $[0, \sigma_u]$. The binary search terminates until the search interval is sufficiently short. The algorithm outputs the best found sanitized output (the last one that was successfully generated; the one with the samllest σ).

4.2 Genobf Function

It is a difficult problem to find (k, ϵ)-*obf* sanitized solutions using a given parameter σ over the intractable search space. The function `genObf` separate the search process into running two independent modules: (1) uncertain noise generative models and (2) privacy tests. The first constructs a utility-preserving noise generative model. By contrast, the privacy test aims to safeguard the privacy of generated obfuscation. It utilizes the random search to alleviate the combinational intractability. Multiple randomized attempts are performed. Every obfuscated output is subject to this privacy test. Iff all the attempts fail, `genObf` returns failure sign. Otherwise, it returns the found (k, ϵ)-*obf* instance.

Each construction attempt begins with selecting a subset of edges subject to alteration. Then, it assigns the deviation among selected edges and injects uncertainty. While, the randomization process heavily relies on the heuristic. In particular, Obf suggests calibrating the perturbation applied to an edge e according to the "uniqueness" of the two nodes u and v. In brief, if both u and v are common nodes *w.r.t.* the property, then r_e should be very small; on the other hand, if u and v are outliers, then r_e should be higher. Meanwhile, edges need to be sampled with the higher probability if they are adjacent to outliers.

4.3 Limitations

The Obf method achieves the desired level of obfuscation with thesmall change in the graph data, thus maintaining high utility. However, this method has two critical weaknesses in the probabilistic context: (1) The design heavily tailored towards the deterministic scenario *e.g.* it assumes the existence of edges is binary (0,1). Thus, it fails to handle uncertain graphs where the existence of edges is probabilistic. All the operators, including edge selection and alteration, need to be integrated with possible world semantics carefully. (2) Its scheme does not consider the structural relevance of edges in critical edge selection/alteration steps, which leads to unnecessary structural distortion. We are left asking the following questions, *how to generalize existing methods to the probabilistic context?* and *how to get a better trade-off between privacy and utility in the probabilistic context?*

5. PRIVACY VIA SQUID

In this section we describe our algorithm, Squid, which injects uncertainty to the given uncertain graph so that it becomes (k, ϵ)-*obf* while preserving as much the stochastic nature as possible. A key feature of our method is to seamlessly integrate edge uncertainty and possible world semantics into the core of anonymization operators.

5.1 Heuristic for Edge Perturbation

Edge selection is often considered as the most central operation in graph anonymization, as well as the most complex one. It needs to balances privacy gain and structural distortion. For the graph, it often needs to consider of the exponential number of edge combinations. Recently, the most

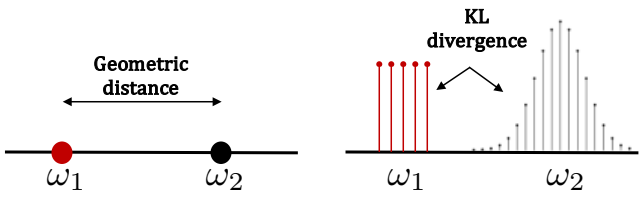


Figure 6: The generalization of uniqueness.

popular paradigm for solving such problems has been using a class of heuristics.

Successes of this approach include (1) anonymity-aware ones that suggest injecting more considerable noise to unique nodes [4, 14, 34] (2) utility-aware ones that avoid distortion over bridge edges whose deletion/addition would significantly impact the graph structure [22,31]. The judicious edge selection must involve two types of heuristics which complement each other. Individually, they are far less effective. However, they have not been explored yet in the context of uncertain graphs.

Motived by the above, we first generalize the calibration of uniqueness with the marriage of KL-divergence function. Second, we propose a generalized version of edge relevance from an information-theoretic perspective. Besides, we develop an efficient algorithm for its evaluation. And, we show the use of multi-heuristics boosts anonymization efficiently and straightforwardly.

Generalized Uniqueness The uniqueness criterion was used to measure how unique a given node is among all the nodes in the graph *w.r.t.* a specific property P. For a given node v, its uniqueness score is the inverse of the commonness score of its property value $w = P(v)$. And, the commonness score of w amounts to the weighted average distance among all other property values.

However, the conventional method merely formulates node properties as discrete values and relies on the geometric distance function to measure their distance. Thus, it fails to handle uncertain graphs where the property values are probabilistic.

We extend the preliminary version to handle the probabilistic case. We first systematically model uncertain property values in both continuous and discrete domains as continuous and discrete random variables, respectively. Then, we consider the use of probability distributions, which are essential characteristics of uncertain property values, in the measuring similarity between uncertain property values. In this work, we use the well known Kullback-Leibler divergence to measure the distance between random variables with parameterized distributions.

Definition 3. ***Uniqueness Score*** *Let P be a property on the set of nodes V of the input graph , let d be the KL divergence function, and let $\theta > 0$ be a parameter. Then the θ-commonness of the property values ω is $C_\theta(\omega)$ amounts to a weighted average over all other property values ω'. while the corresponding uniqueness is $U_\theta := \frac{1}{C_\theta(\omega)}$.*

In the above definition, the weights decays exponentially as a function of the KL divergence among property values, and the parameter θ determines the decay rate.

$$C_\theta(\omega) = \sum_{u \in V} \Phi_\theta(KL(\omega, P(v)))$$

In this work, we set $\theta = \sigma$ as the injected noise blurs the meta-distribution of property values. It should be noted that the commonness and uniqueness are meaningful only as relative measures.

Generalized Edge Relevance Alteration over a single edge would produce local structural change and send ripples through the rest of the graph. Even with the same amount of deviation, the incurred structural distortion varies on the topological role of the altered edge. Targets at the high utility, we should penalize edge modifications according to the incurred structural distortion. It naturally raises the need of measuring the relevance of edges in the probabilistic context.

There are many potential ways to measure it. Importantly, the metric should be able to capture key properties of uncertain graphs. Inspired by the importance of reliability, we measure the edge relevance of a given edge e as the amount of structural distortion, measured by reliability discrepancy, caused by the unit noise subjects to the edge e, as follow.

$$\begin{aligned}\mathcal{ERR}(e) &= \frac{\Delta(\mathcal{G} + r_e)}{|r_e|} \\ &= \frac{\sum_{u,v} |R_{u,v}(\mathcal{G} + r_e) - R_{u,v}(\mathcal{G})|}{|r_e|}\end{aligned}$$

In the conventional case (deterministic graphs with edge addition and deletion $|r_e| = 1$), it amounts to the connectivity distortion, measured by the deviation of # of connected node pairs. In probabilistic graphs, $\mathcal{ERR}$ is used to generalize this concept by quantifying the stochastic impact of partial edge addition/deletion (r_e lies in the contentious range) over the connectivity of all the possible worlds.

Observation 1. *Let $\mathcal{G}_e$, $\mathcal{G}_{\bar{e}}$ denote two uncertain graphs that are identical to $\mathcal{G}$ with $p(e) = 1$ and $p(e) = 0$ respectively. The reliability relevance of an edge e is a constant and equivalent to the following function.*

$$\mathcal{ERR}(e) = \sum_{u,v} R_{u,v}(\mathcal{G}_e) - \sum_{u,v} R_{u,v}(\mathcal{G}_{\bar{e}}) \tag{2}$$

Observe that, the edge relevance only depends on it topological location. It amounts to the difference of the number of connected pairs between two neighbor uncertain graphs.

Proof Sketch. According to the possible world semantics and factorization rule, we can see that

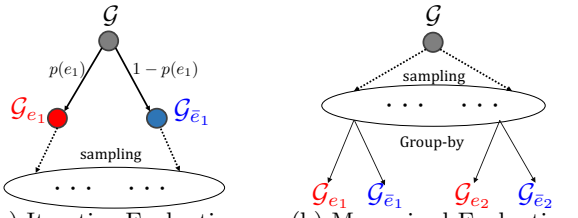
$$R_{u,v}(\mathcal{G}) = p(e) \cdot R_{u,v}(\mathcal{G}_e) \; + \; \big[1 - p(e)\big] \cdot R_{u,v}(\mathcal{G}_{\bar{e}})$$

Note that, the two-terminal reliability $R_{u,v}$ in $\mathcal{G}_e$ and $\mathcal{G}_{\bar{e}}$ are constants. Therefore, two-terminal reliability discrepancy introduced by the single deviation r_e over the uncertain graph $\mathcal{G}$ is equivalent to

$$\begin{aligned}\Delta_{u,v}(\mathcal{G} + r_e) \; &= r_e \cdot R_{u,v}(\mathcal{G}_e) - r_e \cdot R_{u,v}(\mathcal{G}_{\bar{e}}) \\ &= r_e \cdot \; \big[R_{u,v}(\mathcal{G}_e) - R_{u,v}(\mathcal{G}_{\bar{e}})\big]\end{aligned}$$

After aggregation and eliminating the factor r_e, we can see that the reliability relevance of an edge e is equivalent to Equation 2. □

Equation 2 reminds us the basic concept of cut-edge. In a deterministic graph, a cut-edge is an edge of a graph whose deletion increase its number of connected components. We can see that the cut-edge is a binary version of $\mathcal{ERR}$. While $\mathcal{ERR}$ is a continuous function regarding the edge deviation



(a) Iterative Evaluation (b) Memorized Evaluation

Figure 7: Sampling-based reliability detection

r_e and reliability discrepancy. $\mathcal{ERR}$ is not only relevant to connectivity discrepancy but also consider its scale.

Re-visiting the Computation Challenge As shown in Equation 2, the evaluation of $\mathcal{ERR}(e)$ involves a fundamental problem concerning uncertain graphs, which we call the Two-Terminal Reliability detection (TTR) problem. Since this problem is #P-complete, we focus on efficiently and accurately approximate TTR. The Monte-Carlo sampling method can be used to estimate the underlying reliability of an uncertain graph. Namely, we create a subset of possible worlds of the given uncertain graph with the use of edge sampling probabilities. Then, we take the average of the number of connected node pairs in the sampled worlds as an approximation.

It is not trivial to evaluate $\mathcal{ERR}$ over all the edges. One option is to iteratively invoke the sampling-based reliability computation over all the edges, as illustrated in Figure 7(a). It is straightforward to compute the connected components of a graph in linear time (regarding the numbers of the nodes and edges of the graph) using either breadth-first search or depth-first search. For each edge, we need to perform the connected component detection for N sampled graphs. Thus, the overall time complexity is typically in the order of $\mathcal{O}(N|E|^2)$. Apparently, the iterative evaluation is inefficient when the uncertain input graph is massive.

Here, we present an efficient method with the time complexity in the order of $\mathcal{O}(N|E|)$. As illustrated in Figure 7(b), it memories the connected components detection result of samples. For evaluating the reliability relevance of one edge e, we group the sampled possible worlds according to the edge existence, then get the average value of cc over each group as accurate approximation of $cc(\mathcal{G}_e)$ and $cc(\mathcal{G}_{\bar{e}})$. Instead of sampling and evaluation from the scratch, we utilized the memorized results. The running time analysis roughly follows the analysis of the single-edge case. By this way, we bring the evaluation of edge reliability relevance to the realm.

Edge Selection Boosted by Multi-heuristics To get the balance between anonymity and utility, we combined the generalized uniqueness metric U and relevance R metric by taking the ratio between them. We denote the combination as Balance Factor (BF) which is defined as follows:

$$BF(v) = \frac{U(v)}{R(v)}$$

For a give vertex v, $U(v)$ is the normalized uniqueness score of its property value among all the node in the original graph; $R(v)$ is the sum of reliability relevance of edges adjacent to the vertex. The higher value of BF represents the better trade-off between anonymity preserving and utility preserving. Our algorithm uses this heuristic to select and perturb the edge of uncertain graphs, as outlined in Algorithm 2.

5.2 Stochastic Uncertainty Injecting Scheme

For each sampled edge e with the distributed standard deviation σ_e, we select the probability deviation r_e where $r_e \leftarrow R_{\sigma_e}$. Thus, the remaining question is *how can we safely alter edge probability for higher anonymity?* Given the deterministic graph, the idea of uncertainty injecting is to transfer probabilities from existing edges to potential edges, as outlined in Equation 1. However, the uncertainty injecting scheme is unexplored in the uncertain scenario. There

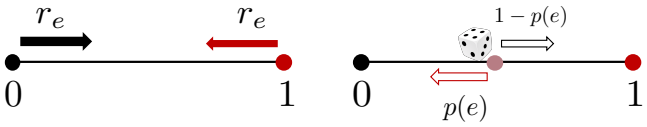


Figure 8: The generalized uncertainty injecting scheme.

are many potential ways to extend the uncertain injecting scheme to the probabilistic context. One option is the Most-Provable-Scheme where the edge probability alteration follows the most probable policy.

$$p(e) = \begin{cases} p(e) - r_e & p(e) \geq 0.5 \\ p(e) + r_e & otherwise \end{cases}$$

Another simple alternative is to consider the probability the edge exists as an indicator of the scheme. It performs the following uncertainty injecting scheme.

$$p(e) = [p(e)] \cdot (1 - r_e) + [1 - p(e)] \cdot r_e \quad (3)$$
$$= p(e) + [1 - 2p(e)] \cdot r_e \quad (4)$$

It has already been used as the uncertainty injecting scheme in our previous work [11]; The flexible value of the edge probability is limited to the range between $p(e)$ and $1-p(e)$. Intuitively, it requires maximizing degree entropy/variance.

However, its connection to the core anonymity objective and the mathematical foundation are unclear. In this work, we show the stochastic uncertainty injecting scheme can maximize the related objective function of anonymity.

The relaxed version of anonymity To simplify the discussion, let us consider the primary case k-obf which low bounds the entropy of posterior-probability distribution over all the nodes. It imposes a set of hard constraints on the output solution. To make it formal, let us define the set of feasible solutions satisfying all the constraints as:

$$\{\mathcal{G} \;\; s.t. \;\; \forall v \; \mathrm{E}(v) \geq \log k\}.$$

$k-$obf can be expressed as joint satisfaction of a set of constraints since the uncertain graph is said to be k-obf iff it $k-$obfuscates all the nodes. Then, in order to sanitize graph, we could maximize the following function:

$$\mathcal{F}_e = \prod_{v \in V} \mathcal{C}_v$$

where

$$\mathcal{C}_v = \begin{cases} 1 & \mathrm{E}_v \geq \log k \\ 0 & otherwise \end{cases}$$

The single constraint $\mathcal{C}$ is either fully satisfied or thoroughly violated. The discontinuity limits the opportunity of

gradient-based optimization methods. The connection between our objective function is still unclear. Thus, we relax the single constraint $\mathcal{C}$ to a fuzzy relation as:

$$\mathcal{C}_v = e^{\mathcal{E}_v - \lambda}.$$

Then, the satisfaction becomes a continuous and differential function *w.r.t.* the entropy. It turns out that the problem can be reduced to maximizing a much simpler function with the relaxed version of anonymity.

Observation 2. *The maximization of $\mathcal{F}_e$ is equivalent to the maximization of the following function:*

$$\mathcal{F} = \sum_{v \in V} E(v) - \|v\| E(\Omega)$$

Targeting at high utility, it aims at increasing the uncertainty at the vertex level $\sum_{v \in V} E(v)$.

Proof Sketch. Let Ω presents the domain of property values in the given graph; we can see that

$$\mathcal{F}_e = \prod_{\omega \in \Omega} \underbrace{\mathcal{C}_\omega \ldots \mathcal{C}_\omega}_{s(\omega)},$$

Taking logarithm for both sides and combining the approximation, our goal is actually to maximized

$$\begin{aligned} \log \mathcal{F}_e &= \sum_{\omega \in \Omega} s(\omega) \ \log C(\omega) \\ &= \sum_{\omega \in \Omega} s(\omega) \big[E(\omega) - \lambda \big] \\ &= \sum_{\omega \in \Omega} s(\omega) E(\omega) \ - \ \|v\| \lambda \end{aligned}$$

Therefore, after removing the constant $\|v\|\lambda$, our goal is actually to maximize $\sum_{\omega \in \Omega} s(\omega) E(\omega)$. It bridges the anonymity of the overall graph $\mathcal{F}_e$ with the component coding (the disorder) of the fuzzy matching matrix.

To perform data coding, we have two angles, row and column, that gain the same result of coding length $\mathcal{L}$ as:

$$\begin{aligned} \mathcal{L} &= \sum_{v \in V} E(v) + \|v\| \log \|v\| \qquad\qquad (row) \\ &= \sum_{\omega \in \Omega} s(\omega) E(\omega) \ + \ \|v\| E(\Omega) \qquad (column). \end{aligned}$$

Therefore, after removing the constant $\|v\| \log \|v\|$, our goal is actually to maximize $\mathcal{F}_e$. □

Observation 3. *Equation 4 approximates the gradient-based exploration of the simplified objective function F.*

Proof Sketch. Fix such node $v \in \mathcal{G}$ and let $e_1, \ldots, e_l$ be l edges that include v. Letting d_v be the random variable corresponding to the degree of v, we have

$$d_v \ = \ \sum_{i}^{l} p(e_i).$$

As ever described, Squid selectively modifies edges connected to unique nodes (*i.e.*, nodes with high degree). The degree of such node v can be approximated by the normal distribution. (The Central Limit Theorem becomes effective already for $l \approx 30$; for unique nodes in the real uncertain graph, the approximation becomes accurate enough.) We have

$$\begin{aligned} d_v \ &\sim \ \mathcal{N}(u, \sigma^2); \\ \sigma^2 \ &= \ \sum_{i=1}^{l} Var(P(e_i)) = \sum_{i=1}^{l} p(e_i)(1 - p(e_i)) \end{aligned}$$

The entropy of the random variable, $E(v)$, can then be approximated by the differential entropy of the normal distribution as $\frac{1}{2}\ln(2\pi\sigma^2) + \frac{1}{2}$. The gradient of our objective function *w.r.t.* the edge existence can be rewritten as

$$\begin{aligned} \nabla \mathcal{F} \ &= \ \nabla E_v \ = \ \nabla \ln(\sigma^2) \\ &\sim \frac{1}{\sigma^2} \nabla \sigma^2 \ \sim \ \frac{1}{\sigma^2} \big[1 - 2 \ p(e_i) \big]. \ \square \end{aligned}$$

5.3 The Squid-obf Algorithm

Algorithm 2 Squid-genObf

Input: Uncertain graph $\mathcal{G}$, the desirable obfuscation level k, tolerance parameter ϵ, the standard deviation σ, $t = 5$ five attempts and auto tuned size multiplier c.
Output: A pair of the sanitized output $\mathcal{G}_{\text{obf}}$ and $\hat{\epsilon}$.

1: Compute the generalized uniqueness score U
2: Compute the reliability relevance R
3: Get $\mathcal{H}$: $\frac{\epsilon}{2}\|v\|$ nodes with highest $U(v)R(v)$ score
4: Compute and normalize balance factor $BF(v) = \frac{U(v)}{R(v)}$
5: $\hat{\epsilon} \leftarrow 1$
6: **for** t times **do**
7: **repeat**
8: $E_C \leftarrow E$
9: randomly pick a vertex $u \in V \setminus \mathcal{H}$ according to BF
10: randomly pick a vertex $v \in V \setminus \mathcal{H}$ according to BF
11: **if** $(u, v) \in E$ **then**
12: $E_C \leftarrow E_c \setminus \{(u, v)\}$ with the probability $p(e)$
13: **else**
14: $E_c \leftarrow E_c \cup \{(u, v)\}$
15: **end if**
16: **until** $E_C = c|E|$
17: **for all** $e \in E_C$ **do**
18: Compute $\sigma(e)$
19: Generate $r_e \leftarrow R_{\sigma(e)}$
20: Inject uncertainty: $p(e) \leftarrow p(e) + \big[1 - 2p(e)\big] \cdot r_e$
21: **end for**
22: $\epsilon_s \leftarrow anonymityEvaluation(\mathcal{G}_s)$
23: **if** $\epsilon_s < \min(\hat{\epsilon}, \epsilon)$ **then**
24: $\hat{\epsilon} \leftarrow \epsilon_s$, $\mathcal{G}_{\text{obf}} \leftarrow \mathcal{G}_s$
25: **end if**
26: **end for**
27: return $\langle \hat{\epsilon}, \mathcal{G}_{\text{obf}} \rangle$

The Squid-obf performs following steps to judiciously use the "uncertainty budget" σ.

[Line 1-2] It computes the uniqueness and relevance level for each vertex $v \in V$. The more unique a vertex is, the harder it is to obfuscate it. The more relevant a vertex is, the more significant impact the mutation brings.

[Line 3] It first select the set $\mathcal{H}$ of $\frac{\epsilon}{2}\|v\|$ nodes, and exclude them from the subsequent mutation effort. Instead of excluding the nodes requires most considerable mutations (uniqueness), Squid-obf also excludes the ones with significant structural relevance.

[Line 4] The mutation over edges is needed to obfuscate the set of nodes not in $\mathcal{H}$. Higher uncertainty is necessary t obfuscate "outliers", nodes with higher uniqueness score.

Table 1: Characteristics of datasets and privacy parameters

Name	Content	Nodes	Edges	ϵ
PPI	Protein-protein Interaction	12K	397K	10^{-2}
BK	Location-based OSN	58K	214K	10^{-3}
DBLP	Co-authorship Network	824K	5M	10^{-4}

Thus, edges need to be sampled with the higher probability if they are adjacent to unique nodes. Mutation over "bridge-like" edges need to be minor to preserve data utility. Thus, edges with higher relevance score need to be sampled with lower probability. We utilize two complementary heuristics, uniqueness (the anonymity-aware one) and relevance (the utility-aware one) to guide our edge selection and mutation. In particular, it assigns a probability to every $v \in V$ which is proportional to the balance factor $BF(v)$ of v.

6. EMPIRICAL STUDY

We evaluated the performance of Squid in comparison with the Rep-based Anonymization (RA) and the Casting-based Anonymization (CA) methods.

Recall that the RA method first extracts a single representative deterministic instance via algorithms proposed by Parchas *et al.* in [23]. Then, RA utilizes the deterministic graph anonymization method [4] to produce the sanitized output. While, the CA method first casts the given uncertain graph into a weighted deterministic graph. Then, CA utilizes the weighted graph anonymization method, proposed by Sudipto *et al.* [9] to generate the corresponding sanitized output. Finally, it re-casts the output to the probabilistic version.

6.1 Experiment Settings

Datasets We test three obfuscation methods on three uncertain graphs: PPI, BrightKite (BK) and DBLP as described in Table 1. These datasets have been widely used in the study of uncertain graphs [16, 24, 36].

- PPI is a dataset of protein-protein interactions, provided by Disease Module Identification DREAM Challenge. The probability of any edge corresponds to the confidence that the interaction actually exists, which is obtained through biological experiments.
- BrightKite (BK) is a location-based social network. In this dataset, each node represent a user. The probability of any edge corresponds to the chance that two users visit each other.
- DBLP is a dataset of scientific publications and authors. Each node represent an author. Two authors are connected by an edge if they have co-authored in a project. The uncertainty on the edge denotes the likelihood that the two authors will collaborate in a new project.

Note that edge probability values of BrightKite and DBLP are obtained by prediction models built on historical logs which contain sensitive and confidential information. These graphs not only capture different real-world scenarios but also different data characteristics *e.g.*, size, density, edge probability. The graphs size vary from 12K of PPI, 58K of BK, to 824K of DBLP with different densities; DBLP is the largest but also the sparest dataset. The probability values

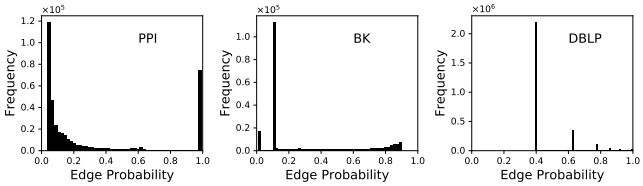


Figure 9: Edge probability distributions.

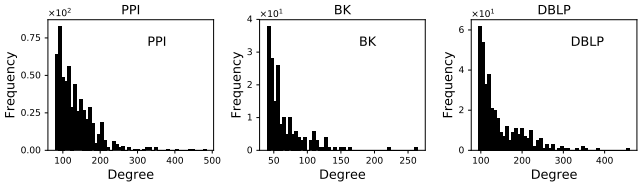


Figure 10: Degree distributions.

of PPI and BrightKite are generally very small. The PPI dataset has a more uniform probability distribution. While, DBLP dataset only has a few probability values.

We report their degree distributions of unique nodes whose obfuscation level is smaller than 300 in Figure 10. Observe that, all the three graphs have a heavy-tailed degree distribution. Namely, they are difficult to be obfuscated.

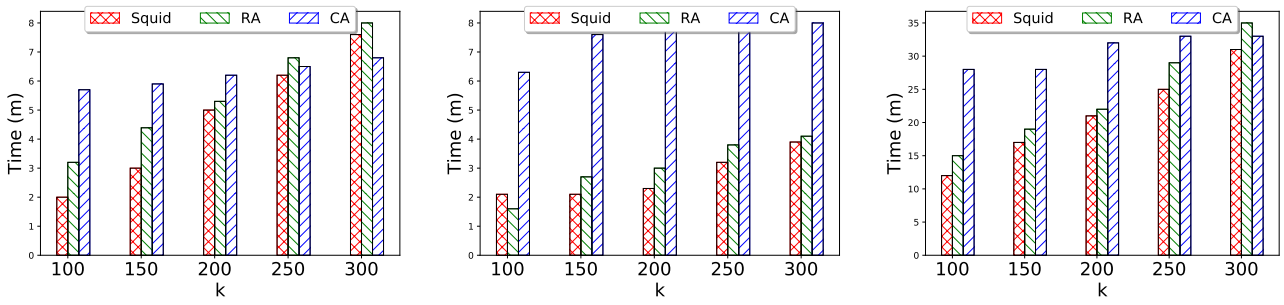
Parameter Setting We consider the obfuscation level k in the range $[100, 300]$, and possible tolerance values ϵ to explore their performance difference.

Statistics For every uncertain graph (obfuscated outputs and original ones), we sampled N possible worlds to compute its statistics of interest. Note that it has been shown that 1000 possible worlds usually suffices to achieve accuracy converge. Then, we compute the discrepancy of obfuscated outputs against the original one. The smaller discrepancy, the better uncertain graph data utility preserving.

6.2 Performance Measures

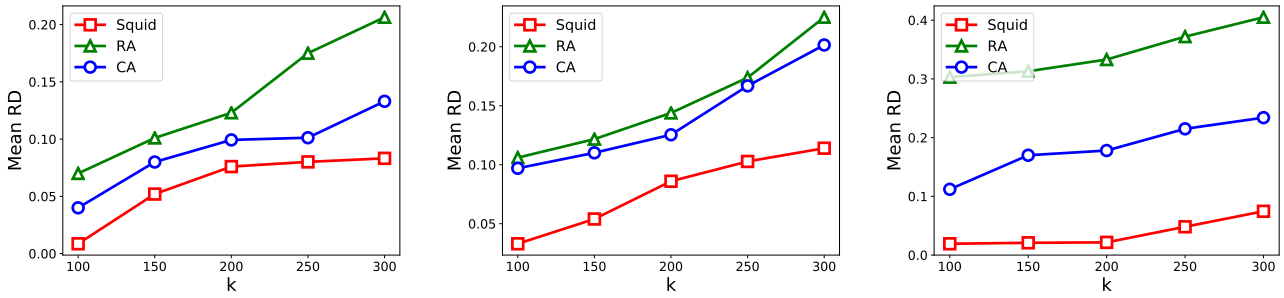
These anonymization algorithms were implemented in C++ and run on Intel Core i7 CPU, 2 GHz, 6MB cache size. We measure the time taken for both the pre-processing (representative extraction step in RA) and the synthetic graph generation. Figure 11 shows the time taken to produce obfuscated results of PPI, BK, DBLP. For larger values for anonymity level k, all the methods take more time to find the sanitized solutions. It is because the increased efforts (more noises & more modified edges) needed to achieve the higher anonymity level (larger values of k). In comparison, our Squid achieves the comparable efficiency with RA and CA methods.

As shown in Figure 11, the time efficiency of our Squid is close to RA, much better than CA. Squid and RA use the same randomized search strategy to identify sanitized solutions using the given standard deviation σ, while Squid adopts a connectivity model in the probabilistic graph context instead of the degree sequence model adopted in RA. Thus, time efficiency depends on how easy it is to find obfuscated solutions that pass the privacy test. Note that, the computation of heuristics (generalized uniqueness and reliability relevance) can be finished off-line. In the same time, multi-heuristics can co-boost the randomized search strategy in the quite straightforward way with little extra effort. Even for severe privacy parameter (*e.g.*, $k = 300$), Squid can



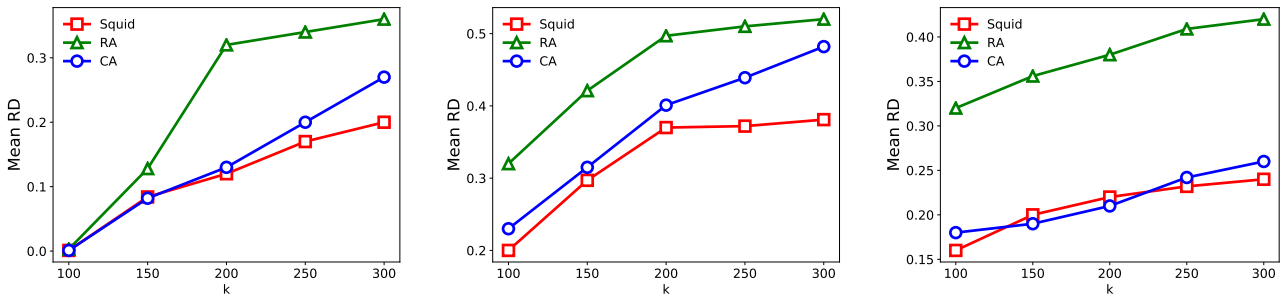
(a) PPI (b) BK (c) DBLP

Figure 11: Time efficiency of different anonymization methods on three real graphs.



(a) PPI (b) BK (c) DBLP

Figure 12: Reliability Discrepancy (RD) of sanitized output graphs of different anonymization methods against original graphs for various values of obfuscation parameter k. The smaller the discrepancy the more information of connectivity structure is preserved.



(a) PPI (b) BK (c) DBLP

Figure 13: Relative error of Shortest Path Distance (SPD) of sanitized output graphs of different anonymization methods against original graphs for various values of obfuscation parameter k.

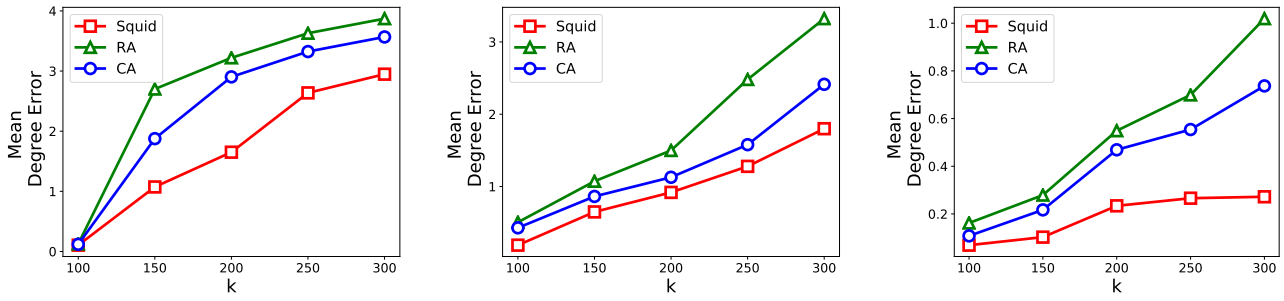
efficiently find obfuscated solutions. Meanwhile, CA builds a linear programming (LP) model which preserves properties of weighted graphs are expressible as linear functions of the edge weights. It uses LP solver to find solutions where the computed solutions constitute the weights in the obfuscated graph.

6.3 Uncertain Graph Statistic Preserving

In the set of experiment, we focus on evaluating their performance concerning statistic preserving. The evaluation includes two groups of graph statistics. The first group includes node separation statistics that quantify the interconnectivity and density of the overall graph. This group includes metrics such as Shortest Path Length [2],Reliability and Graph Diameter. The second group includes degree-based statistics such as Average Node Degree and Degree Distribution. These topological statistics that characterize how degrees are distributed among nodes. Our results are highly consistent across our pool of graph statistics. For brevity, we report only Reliability, Shortest Path Length and Node Degree as their representatives.

Node Separation Statistic. We plot the connectivity distortion introduced by anonymity process in Figure 13. As expected, larger k introduced more significant connectivity

[2]In particular, we use Hyper ANF [5] to approximate shortest path-based statistics.



(a) PPI (b) BK (c) DBLP

Figure 14: Degree error of sanitized uncertain graphs of different anonymization methods on three real graphs for various values of obfuscation parameter k. The smaller the degree error the more information of degree distribution is preserved.

distortion, because more noise was added to achieve the desired level of anonymity. These observations are consistent across biological and social uncertain graphs.

There is a clear improvement of Squid over CA and RA methods for the integration of possible world semantics and uncertain-aware search strategy. For example, in all the dataset ($k = 300$), the reliability discrepancy introduced by Squid is well below 0.1 whereas the ones added by CA is below 0.2. The reliability discrepancy introduced by RA on PPI, BK, DBLP is around 0.2, 0.2, 0.4 respectively. We also observe that as the size of graph increases (PPI $\rightarrow$ DBLP), the performance gap becomes larger and larger.

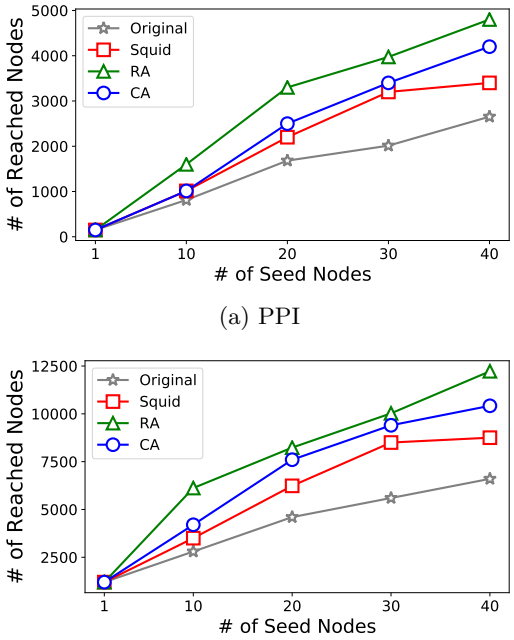
Note that CA and RA schemes deteriorate data utility due to the disregarding the possible world semantics. For example, in the case, $k = 100$ (weak privacy guarantee required little noise), CA and RA incur relatively large connectivity distortion. The representative extraction step of RA introduces noise and results in cumulative errors in the anonymization step. Consequently, sanitized results differ from the original ones. The CA scheme fails to reflect the connectivity of uncertain graph correctly. Therefore, it produces inferior results even with the focused anonymization strategy.

Degree-based Statistic. Figure 14 shows the error of Node Degree values on PPI, BK and DBLP compared to their sanitized outputs. The obfuscated output of Squid, CA capture well Node Degree in all datasets; our method Squid is consistently better. In the largest dataset DBLP, the degree error 0.2 is much lower than 1.08 imposed by the two-phase anonymization scheme (RA) in the same level of obfuscation ($k = 300$). As with previous experiments, the performance gap increases as the graph size increases.

Summary. Our experimental results show that sanitized outputs generated by Squid exhibit structural features close to those of their original uncertain graphs. Results show that we are able to effectively balance the utility and privacy in the probabilistic graph context. The result is encouraging because we can eliminate the noise by moving from the reliability model to a more accurate graph model incorporating with the possible world semantics.

6.4 Uncertain Graph Application Measures

For a sanitized graph to be meaningful for research use, it should produce the close results as the original graph in the application-level settings. Influence Maximization is known as essential techniques with advertisement and public relation campaigns. It tries to locate users in the network who can most quickly spread information through the network. Usually, related algorithms start with identifying the nodes which can maximize influence and model the spread of influence through network how many users has ultimately reached. Motivated by these facts, we compare the result of influence maximization on sanitized outputs and original graphs. In general, it enables us to quantify the trade-off between application utility and privacy protection.



(a) PPI

(b) BK

Figure 15: Results of the influence maximization algorithm on PPI and BK datasets, compared to sanitized outputs where k=100.

Methodology. We used the sampling method to get the expectation value of influence. For each sampled graph, we adopt the degree discount method (a heuristic-based method with light computation) to find the most influential nodes (seeds) on a deterministic graph. Starting from those seeds, we run the weighted cascade influence dissemination model to determine the number of reached users in the network.

For the limit of computation resource (memory overhead), we only perform the set of experiment over PPI and BK

datasets. We report the expected number of reached nodes when the number of initial seeds in Figure 15. There are clear and visible trends across PPI and BK datasets. Graphs with privacy protection diverge from the results of original graphs. We found that our method excelled in preserving the information propagation pattern, followed by CA that aim explicitly at preserving linear property. Overall, our experimental assessment on influence maximization applications confirms our intuition: by incorporating the possible world semantics into the core of anonymization such as edge selection and uncertainty injecting, one can achieve the same desired level of anonymity with a smaller impact in the uncertain graph utility.

7. CONCLUSION

In this work, we first identify the overlooked problemuncertain graph anonymization. We develop a new scheme, Squid, which integrates edge uncertainty into core anonymization steps. It excels in identifying sanitized uncertain graphs with excellent quality. Experiments on three real-world datasets verify its effectiveness and practical utility. There are many potentials to explore further. In real-world graphs, edge probabilities sometimes are not independent, but dependent. We leave the conditional probability model as a future extension. Another extension is to investigate sharing uncertain graphs in the differentially private manner.

8. REFERENCES

[1] E. Adar and C. Re. Managing uncertainty in social networks. *IEEE Data Eng. Bull.*, 2007.

[2] T. M. T. Alina Campan. A clustering approach for data and structural anonymity in social networks. *PinKDD*, 2008.

[3] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava. Class-based graph anonymization for social network data. *VLDB*, 2009.

[4] P. Boldi, F. Bonchi, A. Gionis, and T. Tassa. Injecting uncertainty in graphs for identity obfuscation. *VLDB*, 2012.

[5] P. Boldi, M. Rosa, and S. Vigna. HyperANF: approximating the neighbourhood function of very large graphs on a budget. *CoRR*, 2011.

[6] F. Bonchi, A. Gionis, and T. Tassa. Identity obfuscation in graphs through the information theoretic lens. *ICDE*, 2014.

[7] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. *KDD*, 2011.

[8] Colbourn and Colbourn. The combinatorics of network reliability. 1987.

[9] S. Das, O. Egecioglu, and A. Abbadi. Anonymizing weighted social network graphs. *ICDE*, pages 904–907, 2010.

[10] W.-Y. Day, N. Li, and M. Lyu. Publishing graph degree distribution with node differential privacy. *SIGMOD*, 2016.

[11] X. Dongqing, E. Mohamed Y, and K. Xiangnan. Sharing uncertain graphs using syntactic private graph models. *ICDE*, 2018.

[12] S. Hartung and N. Talmon. The complexity of degree anonymization by graph contractions. *TAMC*, 2015.

[13] M. Hay, G. Miklau, D. Jensen, D. Towsley, and C. Li. Resisting structural re-identification in anonymized social networks. *VLDB*, 2010.

[14] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing social networks. 2007.

[15] M. Hua and J. Pei. Probabilistic path queries in road networks: traffic uncertainty aware path selection. *EDBT*, 2010.

[16] R. Jin, L. Liu, B. Ding, and H. Wang. Distance-constraint reachability computation in uncertain graphs. *VLDB*, 2011.

[17] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. 2003.

[18] J. Lee and C. Clifton. How much is enough? choosing ϵ for differential privacy. *ISC*, 2011.

[19] K. Liu and E. Terzi. Towards identity anonymization on graphs. *SIGMOD*, 2008.

[20] L. Liu, J. Wang, J. Liu, and J. Zhang. Privacy preservation in social networks with sensitive edge weights. *SDM*, 2009.

[21] H. Nguyen, A. Imine, and M. Rusinowitch. Anonymizing social graphs via uncertainty semantics. *CCS*, 2015.

[22] M. Ninggal and J. H. Abawajy. Utility-aware social network graph anonymization. *J Netw Comput Appl*, 2015.

[23] Parchas, Gullo, Papadias, and Bonchi. The pursuit of a good possible world: extracting representative instances of uncertain graphs. *SIGMOD*, 2014.

[24] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios. K-nearest neighbors in uncertain graphs. *VLDB*, 2010.

[25] D. Proserpio, S. Goldberg, and F. McSherry. A workflow for differentially-private graph synthesis. *WOSN*, 2012.

[26] A. Sala, X. Zhao, C. Wilson, and H. Zheng. Sharing graphs using differentially private graph models. *IMC*, 2011.

[27] M. Skarkala, M. Maragoudakis, S. Gritzalis, L. Mitrou, H. Toivonen, and P. Moen. Privacy preservation by k-Anonymization of weighted social networks. *ASONAM*, 2012.

[28] L. Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 2002.

[29] T. Tassa and D. J. Cohen. Anonymization of centralized and distributed social networks by sequential clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2013.

[30] Y. Wang, X. Wu, and L. Wu. Differential privacy preserving spectral graph analysis. *PAKDD*, 2013.

[31] Y. Wang, L. Xie, B. Zheng, and K. C. K. Lee. Utility-oriented k-anonymization on social networks. *DASFAA*, 2011.

[32] W. Wu, Y. Xiao, W. Wang, Z. He, and Z. Wang. k-symmetry model for identity anonymization in social networks. *EDBT*, 2010.

[33] Q. Xiao, R. Chen, and K. Tan. Differentially private network data release via structural inference. *KDD*, 2014.

[34] X. Ying, K. Pan, X. Wu, and L. Guo. Comparisons of randomization and k-degree anonymization schemes

for privacy preserving social network publishing. *SNA-KDD*, 2009.

[35] X. Ying and X. Wu. Randomizing social networks: a spectrum preserving approach. *SIAM*, 2008.

[36] B. Zhao, J. Wang, M. Li, F. Wu, and Y. Pan. Detecting protein complexes based on uncertain graph model. *TCBB*, 2014.

[37] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. *ICDE*, 2008.

[38] L. Zou, L. Chen, and M. T. Özsu. K-automorphism: A general framework for privacy preserving network publication. *Proc. VLDB Endow.*, 2009.